Dnsseczonetoolその後

藤原和典
<fujiwara@wide.ad.jp>

dnsseczonetool

- 作成の動機
  - dnssec-keygen, dnssec-signzoneはよいツールだが、鍵を覚えておくのが面倒
  - 再署名も面倒
  - BIND 9.7のDNSSEC for Humansはやりすぎ
    - keygenで、毎回いつまでの鍵か指定するのはめんどー
    - 古い鍵ファイルを消してくれない
  - 鍵番号などを適度に管理してくれ、signとかrolloverとするだけで動くようなwrapper scriptがほしい
- 公開場所
  - http://member.wide.ad.jp/~fujiwara/dnssec/
  - 使い方などを追記予定ですが、すすんでません

使い方

- dnsseczonetoolを適当なディレクトリにコピー
  - /etc/namedb/master/ など
- dnsseczonetool.confを作成
- dnsseczonetool keygen <domainname>で鍵生成
- ゾーンファイルを、ゾーン名に一致させること
- dnsseczonetool sign <domainname>で署名
- named.confに指定するゾーンファイル名は、ゾーン名.signedとする
- cronにてdnsseczonetool sign <domainname>を実行するだけで定期再署名

dnsseczonetool.conf

ZSK_PARAM="-a RSASHA256 -n zone -b 2048"
KSK_PARAM="-a RSASHA256 -n zone -b 2048 -f KSK“
# 必要があれば –r /dev/urandomを追加

SIGN_PARAM=“-N unixtime”
# NSEC3にしたいときはオプションを追加する

RNDC_OPTION=“-k /etc/rndc.key” など
# rndc reloadのオプションを指定する

ZONELIST=“example.jp”
# 記述するとゾーン名を省略できる

以前からの変更

- ゾーンファイルに一行追加することにしていたのをやめました
  - $INCLUDE “ゾーン名.keys”
  - ldns-signzoneが$INCLUDE非対応のため
  - dnssec-signzoneは二重でもエラーを出さない

- 鍵生成の機能変更
  - keygen2でKSK*2, ZSK*2としていたが、Rolloverの再設計により、keygen, add-next-ksk, add-next-zskを推奨

機能追加

- KSK Rollover
  - 二重署名
  - 鍵追加: add-next-ksk
  - ロールオーバー: ksk-rollover
- ZSK Rollover
  - 事前公開のみから、事前・事後公開に変更
  - RFC 4641によると前のZSKをTTL時間残す必要あり
  - 実際に検証エラーを確認した
  - 鍵追加: add-next-zsk
  - 事後公開の鍵削除: remove-previouskey
- LDNS対応
  - BIND 9を使わないDNSSECの実現のため

KSK Rollover

- 定常的にはKSK 1つ
- 変更時に次のKSKを追加して署名
  - dnsseczonetool add-next-ksk <domainname>
  - dnsseczonetool status <domainname> でDSを表示してレジストリ登録を行うこと
- Rollover時に、従来のKSKを削除して署名
  - dnsseczonetool ksk-rollover <domainname>

ZSK Rollover

1. ZSKは通常は一つ
2. 変更時に次のZSKを追加して署名 → ZSK 2つ
   - dnsseczonetool add-next-zsk <domainname>
3. TTL時間待ったあと
4. ZSK Rollover
   - 従来の鍵を、previous-keyとしてゾーンに維持
   - 次のZSKを現行ZSKにして署名
5. TTL時間待ったあと
6. 過去のZSKの削除と署名 → ZSK 1つへ
   - dnsseczonetool remove-previouskey <domainname>

ZSK Rollover (2)

- TTL時間待つのは面倒: crontab 1行ですませたい
- ZSK常時3つのモデル: packetは太る
  - Next: Rollover後に使用するZSK
  - Current: 現行のZSK
  - Previous: 一つ前に使用していたZSK
- まず、ZSK 2個にするために以下を実施
  - dnsseczonetool add-next-zsk <domainname>
- 定期的に以下を実施してZSK Rollover
  - dnsseczonetool zskroll <domainname>
  - Current ZSKをPrevious ZSKとしてゾーンに保持 (事後公開)
  - Next ZSKをCurrent ZSKに変更して署名に使用
  - Next ZSKを作成してゾーンに公開 (事前公開)

dnsseczonetool まとめ

- Shell scriptだが行数が増えてスパゲティに
  - wc dnsseczonetool
  - 504 1712 13539 dnsseczonetool
  - 2割以上コメントです
- 某所のDNSSEC技術検証で使用
- 10ヶ月以上使用しているが、困ってはいない
  - 困ったら機能追加
- フィードバックありません、、、、
  - 使ってる人いないのかなあ、、、、

オランダ製ソフトウェアによる DNSSEC遊び

藤原和典
<fujiwara@jprs.co.jp>
株式会社日本レジストリサービス

使用するソフトウェアと導入環境

- NL NetLabs製ソフトウェア （ http://www.nlnetlabs.nl/)
  - LDNS　　Cで書かれたDNSライブラリ
  - NSD　　権威DNSサーバ
  - Unbound　　検証サーバ　　http://www.unbound.net/
- Dnsseczonetool
  - keygen, signzoneなどのコマンドを楽に使うスクリプト
  - もともとはBIND 9向けに作成
  - http://member.wide.ad.jp/~fujiwara/dnssec/
- 導入した環境
  - 某ISPの月490円のVPS
  - CentOS 5.5　i686 (32bit)

ソフトウェアのinstall

- CentOSにはNSDなどのパッケージなし
  - BIND 9も古い　9.3.6-4.P1.el5_4.2てなに？
- tar ballを展開
- configure --with-ssl; make; make install
- LDNSは、exampleディレクトリ内もinstallすること
  - cd example
  - ./configure –sysconfdir=/etc/nsd –with-ssl
  - make; make install
  - drillディレクトリでもconfigure; makeするとなおよい
- ldconfig /usr/local/lib か rebootが必要かも
  - unboundはLDNSを使用するため

ldns examples

- LDNSはCで書かれたDNSライブラリで、使用例のプログラムが同梱
- ldns-keygen
  - BIND 9のdnssec-keygenに対応
  - 若干オプションが違う:dnsseczonetool変更不要
- ldns-signzone
  - BIND 9のdnssec-signzoneに対応
  - かなりオプションが違う: dnsseczonetool要変更
  - 機能が低い
    - $INCLUDE, $INCLUDE, $TTLなし
    - -N unixtime なし (SERIALをunixtimeにする)
    - マルチスレッド非対応 (CPU増やしても性能伸びない)
- ldns-key2ds
  - dnssec-dsfromkeyに対応
  - 若干動作が違う: dnsseczonetool要変更

dnsseczonetool

- LDNS対応のため若干変更しました(後述)
- /etc/nsd に dnsseczonetool をコピー
  - chmod +x dnsseczonetool
- dnsseczonetool.conf にldns向けの記述追加

dnsseczonetool.conf (1)

```
# コマンドの読み替え
keygen="/usr/local/bin/ldns-keygen"
signzone="/usr/local/bin/ldns-signzone"
dsfromkey="/usr/local/bin/ldns-key2ds -n"

# BIND 9だとKSK生成は -f KSK だが、ldns-keygenでは -k
# CentOSは/dev/random遅すぎるのでurandom
ZSK_PARAM="-a RSASHA256 -b 2048 -r /dev/urandom"
KSK_PARAM="-a RSASHA256 -b 2048 -k -r /dev/urandom “

# LDNSは-AなしだとZSKによるDNSKEYの署名を作成しない
SIGN_PARAM="-A “

MASTERDIR="/etc/nsd “
# NSDのデフォルトは/etc/nsd
```

dnsseczonetool.conf

```
# RNDCはないので起動させない
RNDC_OPTION=“OFF”

# ldns-signzoneには-N unixtimeの機能がないので、ゾーンファイル
   に_SERIAL_とかき、プリプロセスする
UNIXTIME=`date +%s`
ZONE_PREPROCESS=″sed s/_SERIAL_/$UNIXTIME/ “

# 処理後にreloadさせるコマンドを呼び出す機能を追加
RELOADALL_COMMAND=″/usr/local/sbin/nsdc rebuild &&
   /usr/local/sbin/nsdc reload && echo reloaded “

# その他のオプションは同じ
ZONELIST=″fujiwara.asia secure.crisp.jp badsig.crisp.jp”
```

ゾーン情報の作成

- 通常のゾーンファイルをゾーン名で/etc/nsdに作成
  - 例: /etc/nsd/fujiwara.asia
- シリアル番号を _SERIAL_ に変更
  - 例: fujiwara.asia. IN SOA h.fujiwara.asia. postmaster.fujiwara.asia. (_SERIAL_ 1H 15M 14D 15M)
- 鍵生成
  - /etc/nsd/dnsseczonetool keygen fujiwara.asia
  - /etc/nsd/config/ 以下に鍵が生成され、dnsseczonetoolが管理
- 署名
  - /etc/nsd/dnsseczonetool sign fujiwara.asia
  - ゾーンファイルをプリプロセスし、公開鍵をくっつけ、署名
  - /etc/nsd/ゾーン名.signed が生成される

NSDの設定

- /etc/nsd/nsd.conf を作成

```
server:
        ip-address: 192.0.2.1   # IPアドレス指定

zone:
        name: “fujiwara.asia”
        zonefile: “fujiwara.asia.signed”  # /etc/nsd
        notify: 192.0.2.2         # ゾーン転送する場合
        provide-xfr: 192.0.2.2 NOKEY
```

unboundの設定

- /usr/local/etc/unbound/unbound.conf の作成
- Rootを信用する場合
  - auto-trust-anchor-file, trust-anchorなどの指定
  - root.key: . DS 19036 8 2 49AAC11D7B6F6446702E54A1607
    371607A1A41855200FD2CE1CDDE32 F24E8FB5
- DLVを信用する例
  - dlv.isc.org.keyを/usr/local/etc/unbound/にコピー
  - http://ftp.isc.org/www/dlv/dlv.isc.org.key
  - dlv-anchor-fileを指定

Server:

interface: 127.0.0.1

trust-anchor-file: “root.key”

dlv-anchor-file: “dlv.isc.org.key”

Daemonの起動と使用

- /etc/rc.localに
  - /usr/local/sbin/nsdc start
  - /usr/local/sbin/unbound
  - みなさまはrc.dをきちんと設定してください
- crontab
  - 2 2 * * * /etc/nsd/dnsseczonetool sign
  - 毎日再署名 (zskrollでもよいかな)
- ゾーンの編集時
  - ゾーンファイルの変更後はdnsseczonetoolで署名すること
  - /etc/nsd/dnsseczonetool sign <ゾーン名>
- resolv.conf
  - nameserver 127.0.0.1とかいたが、rebootでもどされた
    - rc.localに echo “nameserver 127.0.0.1” > /etc/resolv.conf
    - うごいてるからいいか、、、